A Sparse Boosting Algorithm for Regression Problem

Po-Hsuan (Cameron) Chen **Yingfei Wang**

Abstract

abstract

1 Introduction

2 Preliminaries

2.1 Boosting algorithms for classification

2.1.1 Classfication problems

Based on labeled training data, the goal of classification is to assign a label for a new objects. During training, a learning algorithm receives as input a training set of labeled examples called the training examples. The output of the learning algorithm is a prediction rule called a hypothesis, which can be treated as a function that maps instances to labels. In classification problems, we assume that the examples are chosen randomly from the same but unknown distribution $\mathcal{D}$.

To measure the quality of a given classifier, we use its error rate. The fraction of mistakes on the training set is called the training error, denoted as $\widehat{err}$. The generalization error of a classifier measures the probability of misclassifying a random example from the distribution $\mathcal{D}$. Generalization error of a hypothesis h is denoted as $err_{\mathcal{D}}(h)$.

2.1.2 Boosting

A boosting algorithm is an algorithm that converts a weak learning algorithm into a strong learning algorithm. Like any learning algorithm, a boosting algorithm takes as input a set of training examples $(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)$ from unknown distribution $\mathcal{D}$, where each x_i is an instance from the instance space $\mathcal{X}$ and each y_i is the associated label. We assume that there are only two classes, -1 and $+1$. It has access to a weak learner A which for $\forall D$, given examples drawn from D, computes $h \in \mathcal{H}$ (the hypothesis space of the weak learner) such that $Pr[err_D(h) \leq \frac{1}{2} - \gamma] \geq 1 - \delta$. The goal of a boosting algorithm is to find h with $err_{\mathcal{D}} \leq \epsilon$.

AdaBoost is one of the most famous boosting algorithms. The pseudocode is shown below.

Algorithm 1: AdaBoost

input : m examples $(x_1, y_1) \dots (x_m, y_m)$ where $x_i \in \mathcal{X}$ $y_i \in \{-1, +1\}$

Initialize: $D_1(i) = 1/m, \forall i = 1, ..., m$

For $t = 1, 2, ..., T$:

- construct D_t.
- run A on D_t get weak hypothesis h_t.
- $\epsilon_t = err_{D_t}(h_t) = \frac{1}{2} - \gamma_t$.
- $\alpha_t = \frac{1}{2} \ln(\frac{1-\epsilon_t}{\epsilon_t}) > 0$.
- update, for $i = 1, 2, ..., m$:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \{ \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} .$$

output:

$$H : H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x) \right)$$

2.2 Sparse Representation

Consider a linear prediction model that the target composite predictor is a sparse combination of a set of weak predictor. The sparse representation is interested in identifying those basic weak predictor. We are interested in the case that the number of weak predictor, such as linear functions, step functions, quadratic functions, etc, is larger then the number of training data we have. In this case, an unconstraint error minimization is inadequate because under the scenario of aggregating the weak predictor, the composite predictor is likely to overfit the data. Therefore, the common practice is to impose a sparsity constraint on the weight $\boldsymbol{\alpha}$ to obtained a regularized problem. We define $\|\boldsymbol{\alpha}\|_0 = \{i : \alpha_i \neq 0\}$. The constraint will be $\|\boldsymbol{\alpha}\|_0 \leq k$, where k is a predefined parameter based on our a priori understanding of the problem. A sparse representation of the composite predictor is easier to store, compute and interpret. More over, a simpler predictor is more likely to generalize well to different input data and get a higher prediction accuracy.

3 Gradient Boosting

The goal of regression is to find a function $F(x) \in \{\sum_{i=1}^{T} \alpha_i h_i(x) + const \big| h \in \mathcal{H}\}$ to approximate the true value $y = c(x)$ so as to minimize the expected value of some specified loss function L(y, F(x)). Here $\mathcal{H}$ is called weak hypothesis space, training and test examples are i.i.d. from the same yet unknown distribution $\mathcal{D}$ and y is real valued.

Gradient Boosting tries to find an approximation $\hat{F}(x)$ that minimizes the average value of the loss function on the training set. It does so by starting with a model, consisting of a constant function $F_0(x)$, and incrementally expanding it in a greedy fashion:

$$F_0(x) = \arg\min_{\alpha} \sum_{i=1}^{m} L(y_i, \alpha)$$

$$F_t(x) = F_{t-1}(x) + \arg\min_{h \in \mathcal{H}} \sum_{i=1}^{m} L(y_i, F_{t-1}(x_i) + h(x_i)).$$

However, there is no simply way to exactly solve the problem of choosing at each step the best h for an arbitrary loss function L, so instead, steepest descent is used. If we only cared about predictions at the points of the training set, and h were unrestricted, we can view $L(y, F)$ not as a functional of F, but as a function of a vector of values $(F(x_1), F(x_2), ..., F(x_m))$. And then we can calculate the

gradient of L: $r_{i,t} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{t-1}(x)}$ which is referred to as pseudo-residuals and find step size that minimizes the loss along the negative gradient direction. But as h must come from a restricted class $\mathcal{H}$ of functions we'll just choose the one that most closely approximates the gradient of L (that's what "fit a weaker learner $h_t(x)$ to pseudo-residuals" in gradient boosting pseudocode is doing).

Algorithm 2: Gradient Boosting

input : m examples $(x_1, y_1) \dots (x_m, y_m)$, weak learning algorithm A

$F_0(x) = \text{argmin}_\alpha \sum_{i=1}^{m} L(y_i, \alpha)$
for $t = 1$ *to* T **do**
  pseudo-residuals $r_{i,t} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{t-1}(x)}$, $\forall i = 1, \dots, m$
  fit a weaker learner $h_t(x)$ to pseudo-residuals, i.e. train it using the training set$\{(x_i, r_{i,t})\}_{i=1}^{m}$
  $\alpha_t = \text{argmin}_\alpha \Sigma_{i=1}^{m} L(y_i, F_{t-1}(x_i) + \alpha h_t(x_i))$
  $F_t(x) = F_{t-1}(x) + \alpha_t h_t(x)$
end
output: $F_T(x)$

Loss Function

$$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{i=1}^{m} L(y_i - \sum_{t=1}^{T} \alpha_t h_t(x_i))$$

How do we interpret the final predictor of it?

4 Adaboost Regression

Algorithm 3: Adaboost.RT Algorithm

input : m examples $(x_1, y_1) \dots (x_m, y_m)$, weak learning algorithm A, threshold ϕ

$\forall i,\ D_t(i) = \frac{1}{m}$
$\epsilon_t = 0$
for $t = 1$ *to* T **do**
  run A on D_t get f_t
  are$_t(i) = |\frac{f_t(x_i)}{y_i}|$
  $\epsilon_t = \sum_{i:are_t(i)>\phi} D_t(i)$
  $\beta_t = \epsilon_t^n$, where n =power coefficient
  $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times XXXXX$
end
output: final predictor $F(x) = \frac{\sum_t (log \frac{1}{\beta_t} f_t(x))}{\sum_t (log \frac{1}{\beta_t})}$

What's the loss function of AdaBoost.RT?

How do we interpret the final predictor of it?

5 Adding Sparsity Constraint

The output of boosting algorithm is a linear prediction model. Based on the structure of boosting algorithm, it will keep adding weak predictor into the final predictor to minimize the prediction error. In addition to prediction accuracy, sparsity of the composite predictor is a desirable characteristic. The sparsity over here means that there are only relative small amount of nonzero weight assigned to the composite predictor. Although Adaboost is already known to be less prone to overfit, we hope to derive some property by imposing sparsity constraint on the composite predictor.

6 Sparse Gradient Boosting Regression

Our goal is to solve the regularized loss minimization problem:

$$\min_{\boldsymbol{\alpha}} \mathcal{L}(\boldsymbol{\alpha}) = \sum_{i=1}^{m} L(y_i, \sum_{j=1}^{|\mathcal{H}|} \alpha_j \frac{h_j(x_i)}{\sum_{i=1}^{m} h_j^2(x_i)})$$

$$s.t. \|\boldsymbol{\alpha}\|_1 = \sum_{j=1}^{|\mathcal{H}|} |\alpha_j| \leq C$$

Here, $\sum_{i=1}^{m} h_j^2(x_i)$ can be treat as $\|h_j\|_2$ on training data and $\frac{h_j(x_i)}{\sum_{i=1}^{m} h_j^2(x_i)}$ is a normalized version of weak hypothesis. l_1 regulization on the coefficient of normalized weak hypotheses is more meaningful than simply on the coefficient of original hypotheses.

The idea is to first run Gradient Boosting on training examples until the coefficients α_j's hit the boundary of the above regularization constant and then use "coordinate descent" procedure to solve the constrained optimization problem. To be more specific, at some round t (on which $F_{t-1}(x_i) = \sum_{j=1}^{|\mathcal{H}|} \alpha_j \frac{h_j(x_i)}{\sum_{i=1}^{m} h_j^2(x_i)}$), consider all possible adjustments along two "coordinates": for any two indices j_1, j_2, we can subtract some $a/2$ from coefficient α_{j_1} and add it to α_{j_2} to keep $\sum_{j=1}^{|\mathcal{H}|} |\alpha_j| \leq C$.

Under different loss functions, we can derive different regularized algorithms. For the rest of our work, we choose squared loss as a demonstration. Our proposed algorithm Sparse Gradient Boosting for $L(y, F) = (y - F)^2/2$ is defined as follows:

Algorithm 4: Sparse Gradient Boosting

input : m examples $(x_1, y_1) \ldots (x_m, y_m)$, weak learning algorithm A

$F_0(x) = \text{argmin}_{\alpha} \sum_{i=1}^{m} L(y_i, \alpha)$

for $t = 1$ *to* T **do**

$r_{i,t} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{t-1}(x)}, \ \forall i = 1, \ldots, m$

$h_{t_{max}} = \text{argmax}_{h_j} \frac{\sum_{i=1}^{m} h(x_i) r_{i,t}}{\|h\|_2}$

$h_{t_{min}} = \text{argmin}_{h_j : \alpha_j > 0} \frac{\sum_{i=1}^{m} h(x_i) r_{i,t}}{\|h\|_2}$

$\epsilon = \min(2\alpha_{t_{min}}, \frac{2\sum_{i=1}^{m} r_i (\frac{h_{t_{max}}(x_i)}{\|h_{t_{max}}\|_2} - \frac{h_{t_{min}}(x_i)}{\|h_{t_{min}}\|_2})}{\sum_{i=1}^{m} (\frac{h_{t_{max}}(x_i)}{\|h_{t_{max}}\|_2} - \frac{h_{t_{min}}(x_i)}{\|h_{t_{min}}\|_2})^2})$

$\alpha_{t_{max}} \leftarrow \alpha_{t_{max}} + \frac{\epsilon}{2}$

$\alpha_{t_{min}} \leftarrow \alpha_{t_{min}} - \frac{\epsilon}{2}$

$F_t(x) = \sum_{i=1}^{t} \alpha_i \frac{h_i(x)}{\sum_{i=1}^{m} h_j^2(x_i)}$

end

output: $F_T(x)$

7 Analysis

7.1 Use standard gradient boosting, then using the solution of the standard gradient boosting as warm-start for the sparse gradient boosting

(1) α will be at the boundary of the constraint OR (2) α is at the interior of the constraint set If (2) then it means that the composite predictor perfectly fits the data, which might leads to over fitting the data, consider reduce C. Therefore, we assume that after running the standard gradient boosting, $\boldsymbol{\alpha}$ will lie on the boundary of the constraint set.

7.2 How do we prove that the algorithm is solving the above mentioned l_1 constraint optimization problem?

We use $\mathcal{L}_{j_1 \to j_2}(\boldsymbol{\alpha}, a)$ to denote the value of loss function after an adjustment on j_1 and j_2, where $\mathcal{L}_{j_1 \to j_2}(\boldsymbol{\alpha}, a) = \mathcal{L}(\alpha_1, \alpha_2, ..., \alpha_{j_1} + \frac{a}{2}, ..., \alpha_{j_2} - \frac{a}{2}, ..., \alpha_{|\mathcal{H}|})$.

The following theorem shows that in the "coordinate descent" procedure, Sparse Gradient Boosting algorithm chooses the adjustment on two coordinates which gives the largest gradient descent and the step size a is chosen to achieve the minimum along those directions.

Theorem 1. *In each iteration, the choice of k, l, and s satisfy the following properties*

$$(l, k) = \textit{argmax}_{(j_1, j_2): \alpha_{j_1} > 0} \left(- \frac{\partial \mathcal{L}_{j_1 \to j_2}(\boldsymbol{\alpha}, a)}{\partial a} \right)$$

$$\epsilon = \textit{argmin}_{a: a/2 \leq \alpha_l} \mathcal{L}_{l \to k}(\boldsymbol{\alpha}, a)$$

Taking l_2 loss as the loss function,

$$\mathcal{L}_{j_1 \to j_2}(\alpha, a) = \frac{1}{2} \sum_{i=1}^{m} (y_i - \sum_{j=1}^{|\mathcal{H}|} \alpha_j h_j(x_i) + \frac{a}{2} h_{j_1}(x_i) - \frac{a}{2} h_{j_2}(x_i))^2$$

$$-\frac{\partial \mathcal{L}_{j_1 \to j_2}(\alpha, a)}{\partial a} = \sum_{i=1}^{m} \left((y_i - \sum_{j=1}^{|\mathcal{H}|} \alpha_j \frac{h_j(x_i)}{\|h_j\|_2} + \frac{a}{2} \frac{h_{j_1}(x_i)}{\|h_{j_1}\|_2} - \frac{a}{2} \frac{h_{j_2}(x_i)}{\|h_{j_1}\|_2})(\frac{1}{2} \frac{h_{j_2}(x_i)}{\|h_{j_1}\|_2} - \frac{1}{2} \frac{h_{j_1}(x_i)}{\|h_{j_2}\|_2}) \right)$$

Letting $a = 0$

$$-\frac{\partial \mathcal{L}_{j_1 \to j_2}(\alpha, a)}{\partial a}\bigg|_{a=0} = \sum_{i=1}^{m} \left((y_i - \sum_{j=1}^{|\mathcal{H}|} \alpha_j \frac{h_j(x_i)}{\|h_j\|_2})(\frac{1}{2} \frac{h_{j_2}(x_i)}{\|h_{j_1}\|_2} - \frac{1}{2} \frac{h_{j_1}(x_i)}{\|h_{j_1}\|_2}) \right)$$

$$= \frac{1}{2} \frac{\sum_{i=1}^{m} \left(h_{j_2}(x_i)(y_i - \sum_{j=1}^{|\mathcal{H}|} \alpha_j \frac{h_j(x_i)}{\|h_j\|_2}) \right)}{\|h_{j_2}\|_2} - \frac{1}{2} \frac{\sum_{i=1}^{m} \left(h_{j_1}(x_i)(y_i - \sum_{j=1}^{|\mathcal{H}|} \alpha_j \frac{h_j(x_i)}{\|h_j\|_2}) \right)}{\|h_{j_1}\|_2}$$

To maximize this quantity under the constraint $\alpha_{j_1} > 0$, we must choose

$$j_2 = \text{argmax}_j \frac{\sum_{i=1}^{m} \left(h_j(x_i)(y_i - \sum_{k=1}^{|\mathcal{H}|} \alpha_k \frac{h_k(x_i)}{\|h_k\|_2}) \right)}{\|h_j\|_2},$$

$$j_1 = \text{argmin}_{j, \alpha_j > 0} \frac{\sum_{i=1}^{m} \left(h_j(x_i)(y_i - \sum_{k=1}^{|\mathcal{H}|} \alpha_k \frac{h_k(x_i)}{\|h_k\|_2}) \right)}{\|h_j\|_2}.$$

The residual r_i under l_2 loss is:

$$r_i = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x) = F_{t-1}(x)} = (y_i - F(x_i)),$$

where $F(x) = \sum_{j=1}^{|\mathcal{H}|} \alpha_j \frac{h_j(x)}{\|h_j\|_2}$ and hence

$$j_2 = \text{argmax}_j \frac{\sum_{i=1}^{m} h_j(x_i) r_i}{\|h_j\|_2}$$

$$j_1 = \text{argmin}_j \frac{\sum_{i=1}^{m} h_j(x_i) r_i}{\|h_j\|_2}$$

Set $\frac{\partial \mathcal{L}_{j_1 \to j_2}(\alpha, a)}{\partial a} = 0$, we get:

$$= \sum_{i=1}^{m} \left(\frac{a}{4} (\frac{h_{j_2}(x_i)}{\|h_{j_2}\|_2} - \frac{h_{j_1}(x_i)}{\|h_{j_1}\|_2})^2 - (y_i - \sum_{j=1}^{|\mathcal{H}|} \alpha_j \frac{h_j(x_i)}{\|h_j\|_2})(\frac{1}{2} \frac{h_{j_2}(x_i)}{\|h_{j_2}\|_2} - \frac{1}{2} \frac{h_{j_1}(x_i)}{\|h_{j_1}\|_2}) \right) = 0$$

$$a = \frac{2 \sum_{i=1}^{m} r_i (\frac{h_{j_2}(x_i)}{\|h_{j_2}\|_2} - \frac{h_{j_1}(x_i)}{\|h_{j_1}\|_2})}{\sum_{i=1}^{m} (\frac{h_{j_2}(x_i)}{\|h_{j_2}\|_2} - \frac{h_{j_1}(x_i)}{\|h_{j_1}\|_2})^2}$$

7.3 Compare with gradient boosting

Before we start to run Sparse Gradient Boosting algorithm, we first run Gradient Boosting until the coefficients hit the regularization boundary. Assume at this point, $F(x) = \sum_{j=1}^{\mathcal{H}} \alpha' h_j(x) = \sum_{j=1}^{\mathcal{H}} \alpha \frac{h_j(x)}{\|h_j(x)\|_2}$, with $\sum_{j=1}^{\mathcal{H}} \alpha_j = C$. We then compare the performance of continuing Gradient Boosting and switching to Sparse Gradient Boosting after this point. Unlike Gradient Boosting which only adds weak hypotheses, Sparse Gradient Boosting can also reduce the weight of the worst active hypothesis and transfer the weighted from it to the best one, keeping the coefficients within the regularization boundary (which imposes sparsity on the solution).

Sparse Gradient Boosting is indeed a natural extension of Gradient Boosting. Recall that in Gradient Boosting algorithm, the second step in each iteration t is to fit a weaker learner $h_t(x)$ to pseudo-residual $r_{i,t}$. Different weak learning algorithms, e.g. least-square estimation, will choose different h_t. Here, we choose $h_t = \text{argmax}_h \frac{\sum_{i=1}^m h(x_i) r_{i,t}}{\|h\|_2}$, which is the same as $h_{t_{max}}$ chosen by Sparse Gradient Boosting. Next, we'll first show why choosing weak hypothesis this way is meaningful, and then we'll show that under this weak learner, each iteration step of Gradient Boosting can be viewed as a degenerate case of that of Sparse Gradient Boosting.

First, as we mentioned previously, in Gradient Boosting, we want to choose the weak hypothesis h_t that most closely approximates the gradient of L which is r_t at time t. As before, we view $L(y, F)$ not as a functional of F, but as a vector of values $< F(x_1), F(x_2), ..., F(x_m) >$. Let $\boldsymbol{h} =< h(x_1), h(x_2), ..., h(x_m) >$ and residual vector $\boldsymbol{r}_t =< r_{1,t}, r_{2,t}, ...r_{m,t} >$, then we can rewrite the hypothesis we choose on round t as $h_t = \text{argmax}_h \frac{\sum_{i=1}^m h(x_i) r_{i,t}}{\|h\|_2} = \text{argmax}_h \frac{\boldsymbol{h} \cdot \boldsymbol{r}_t}{\|\boldsymbol{h}\|_2 \|\boldsymbol{r}_t\|_2}$. Note that $\frac{\boldsymbol{h} \cdot \boldsymbol{r}_t}{\|\boldsymbol{h}\|_2 \|\boldsymbol{r}_t\|_2}$ equals the cosine of the angle between these two vectors. The larger this quantity, the smaller the angle, and the closer the two vectors. In particular, if it equals to 1, $\boldsymbol{h}$ is parallel to $\boldsymbol{r}_t$, then as in Gradient Boosting, $\boldsymbol{h}$ is a perfect substitute of $\boldsymbol{r}_t$, which is the negative gradient of L. And here, we want all the α_j's are greater than or equal to 0, so when the angle between $\boldsymbol{h}$ and $\boldsymbol{r}_t$ equal 180^o, it is not preferable.

Next, we'll show that when we choose h_t like this on each round, the Gradient Boosting can be viewed as degenerate case of Sparse Boosting. To see this first recall that after choosing h_t, we choose $\alpha_t = \text{argmin}_\alpha \Sigma_{i=1}^m L(y_i, F_{t-1}(x_i) + \alpha h_t(x_i))$ and then add $\alpha_t h_t(x)$ to current F_{t-1}. Under l_2 loss, we can derive a closed-form expression of α_t. $\Sigma_{i=1}^m L(y_i, F_{t-1}(x_i) + \alpha h_t(x_i)) = \Sigma_{i=1}^m (y_i - F_{t-1}(x_i) - \alpha h_t(x_i))^2 = \sum_{i=1}^m (r_{i,t} - \alpha h_t(x_i))^2$. We take first derivative with respect to α and set it to 0, we get $\alpha_t = \frac{\sum_{i=1}^m r_{i,t} h(x_i)}{\sum_{i=1}^m h^2(x_i)} = \frac{\sum_{i=1}^m r_{i,t} h(x_i)}{\|h_t\|_2}$. Then Gradient Boosting add $\alpha_t h_t(x) = \sum_{i=1}^m r_{i,t} h_t(x_i) \frac{h_t(x)}{\|h_t\|_2}$ to $F_{t-1}(x)$. In Sparse Gradient Boosting, $h_{t_{max}}$ is the same as h_t in Gradient Boosting on the same round. Now if we remove the requirement that $\alpha_{t_{min}} > 0$, which means that $h_{t_{min}}$ is not necessary an active hypothesis, then the worst hypothesis is just the negation of the best one $h_{t_{max}}$ (here we assume the negation of the best h belongs to $\mathcal{H}$). If we allow the coeffecient of the worst hypothesis to be negative, then we can just choose the step size $\epsilon = \frac{2 \sum_{i=1}^m r_{i,t} (\frac{h_{t_{max}}(x_i)}{\|h_{t_{max}}\|_2} + \frac{h_{t_{max}}(x_i)}{\|h_{t_{max}}\|_2})}{\sum_{i=1}^m (\frac{h_{t_{max}}(x_i)}{\|h_{t_{max}}\|_2} + \frac{h_{t_{max}}(x_i)}{\|h_{t_{max}}\|_2})^2} = \frac{\sum_{i=1}^m r_{i,t} h_{t_{max}}(x_i)}{\sum_{i=1}^m h_{t_{max}}^2(x_i)} \|h_{t_{max}}\|_2 = \sum_{i=1}^m r_{i,t} h_{t_{max}}(x_i)$. Then in Sparse Gradient Boosting algorithm, we add $\frac{\epsilon}{2} \frac{h_{t_{max}}(x)}{\|h_{t_{max}}\|_2} = \frac{1}{2} \sum_{i=1}^m r_{i,t} h_{t_{max}}(x_i) \frac{h_{t_{max}}(x)}{\|h_{t_{max}}\|_2}$ and substract $\frac{\epsilon}{2} \frac{h_{t_{min}}(x)}{\|h_{t_{min}}\|_2} = -\frac{\epsilon}{2} \frac{h_{t_{max}}(x)}{\|h_{t_{max}}\|_2}$ to $F_{t-1}(x)$. Since $h_t = h_{t_{max}}$, those two algorithms are adding the same amount of the same weak hypothesis to $F_{t-1}(x)$. That's to say, Gradient Boosting can be viewed as a degenerate case of Sparse Gradient Boosting algorithm.

8 Modified Sparse Gradient Boosting Regression

Algorithm 5: Sparse Gradient Boosting

input : m examples $(x_1, y_1) \dots (x_m, y_m)$, weak learning algorithm A

$F_0(x) = \text{argmin}_\alpha \sum_{i=1}^{m} L(y_i, \alpha)$

for $t = 1$ *to* T **do**

$r_{i,t} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{t-1}(x)}, \ \forall i = 1, \dots, m$

$h_l = \text{argmax}_{h_j} \frac{\sum_{i=1}^{m} h_j(x_i) r_{i,t}}{\|h_j\|_2}$

$h_k = \text{argmin}_{h_j} \frac{\sum_{i=1}^{m} h_j(x_i) r_{i,t}}{\|h_j\|_2}$

$\epsilon = \min(2\alpha_l, \frac{2\sum_{i=1}^{m} r_i(\frac{h_{j_2}(x_i)}{\|h_{j_2}\|_2} - \frac{h_{j_1}(x_i)}{\|h_{j_1}\|_2})}{\sum_{i=1}^{m}(\frac{h_{j_2}(x_i)}{\|h_{j_2}\|_2} - \frac{h_{j_1}(x_i)}{\|h_{j_1}\|_2})^2})$

$\alpha_t \leftarrow \alpha_t + \frac{\epsilon}{2}$

$\alpha_l \leftarrow \alpha_l - \frac{\epsilon}{2}$

$F_t(x) = \sum_{i=1}^{t} \alpha_i \frac{h_i(x)}{\sum_{i=1}^{m} h_j^2(x_i)}$

end

output: $F_T(x)$

9 Experiment

1. iterations vs number of weak predictor
2. iterations vs generalization/training error
3. sparsity parameter k vs generalization error under T iterations + constant line of generation error under T iterations w/o sparsity constraint

10 Conclusion

References

References follow the acknowledgments. Use unnumbered third level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to 'small' (9-point) when listing the references. **Remember that this year you can use a ninth page as long as it contains *only* cited references.**

[1] Zhang, Tong. "Adaptive forward-backward greedy algorithm for sparse learning with linear models." NIPS, 2008.

[2] Shrestha, D. L., and D. P. Solomatine. "Experiments with AdaBoost. RT, an improved boosting scheme for regression." Neural computation 18.7 (2006): 1678-1710.